Real-Time Human Body Tracking in Public Spaces

Honours Project Report, 2004

Kushal Vaghani
kuv10@student.canterbury.ac.nz
Department of Computer Science and Software Engineering
University of Canterbury, Christchurch, New Zealand

Supervisor: Dr. Richard Green

Abstract

Robust tracking and recovery of human body parameters is a vital input to applications such as motion capture, virtual conferencing, surveillance and innovative interfaces supporting gestures. Past approaches for tracking human bodies are either based on infrared sensors, magnetic markers or computer vision. Such a tracking system should be fast enough for real-time and less sensitive to noise. In addition, if the application is placed in a public space such as a cinema hall, there could be additional difficulties. Public spaces are inherently unconstrained environments where clothing, lighting, background, occlusion and reflectance can vary and so represent a significant challenge to tracking techniques. In this research, we develop a novel algorithmic technique that can be used to estimate 3-D (3 dimensional) joint positions for a human body in a public space. This approach is based on markerless vision-based tracking. Our results show that the estimates of the body parameters obtained are sufficiently robust to the changing environment of a public space.

Acknowledgements

First, I would like to deeply thank Dr. Richard Green, my supervisor, for his guidance, supervision and help throughout the year. I would also like to thank Dr. Mark Billinghurst for his incredible support during my honours year. In addition, my sincere thanks to Dr. Ramakrishnan Mukundan for all the discussions on various topics in computer vision and image processing. Finally, thanks to all the postgraduate honours and masters students who made this year very memorable.

List of Publications

K. Vaghani and R. Green. Real-time Motion Capture in Public Spaces Using Stereo Vision. Image and Vision Computing New Zealand (IVCNZ), 21-23 November, Akaroa, 2004.

Contents

List of Figures

List of Tables

1 Introduction

Tracking non-rigid motion from image sequences has been of great interest to the computer vision community. One of the important reasons is because it is very difficult. The problem can be categorised into two different types of motions namely, deformable object motion and motion of an articulated object[6]. Deformable objects are those which alter their shape over time such as a spring. Articulated objects consist of segments held together by joints (figure 1). Human bodies are referred to as articulated objects.

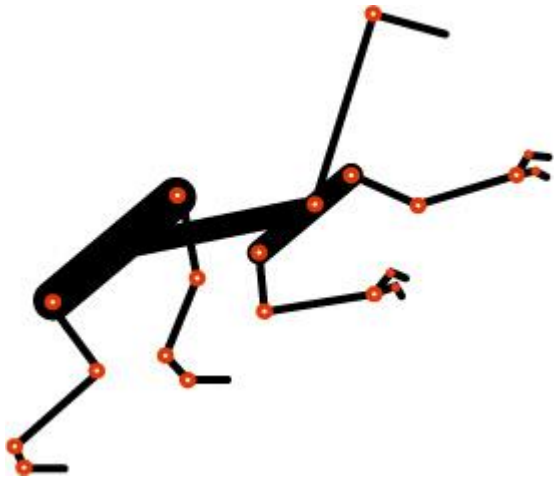

Figure 1: An Articulated Object

Tracking human bodies is very difficult as the geometry of the human body is not well-defined. It could vary significantly from person to person across age, gender and race. Many potential applications which require tracking human bodies operate in public spaces such as a shopping mall or a cinema house (refer figure 2)[1].

An electronic display in a shopping mall, can be controlled by recognizing gestures. An entertainment experience such as motion capture can be placed in a cinema house. The problem to solve in such a situation is tracking human bodies in such an unconstrained environment and identifying various joint positions with varying clothing, lighting, background, occlusion and reflectance.

Previous approaches to human body tracking relied on magnetic or joint markers which could be connected to cables[16]. The subject is supposed to wear a suit with markers on it. A better solution based on marker-free computer vision approach would be less constraining and preferable. As already discussed, there are many other challenges such as noise, occlusion, non-static background, lighting variation and reflectance change. So far, prior computer vision research has been limited to laboratory controlled clothing, lighting, background, occlusion and reflectance[36].

In this research, we develop a novel algorithmic approach to estimate 3-D joint positions of the upper human body in a public space. We will use a motion capture tool called *Lord of the Rings* (section 3.2.2), developed at the Human

[1] http://www.oaaa.org/outdoor/sales/formats/furniture10.asp

Figure 2: A Public Space

Interface Technology Laboratory (HITlab)[2] earlier in this year, as a research platform to evaluate our approach. An overview of our algorithmic technique (explained in chapter 3) is illustrated in figure 3.

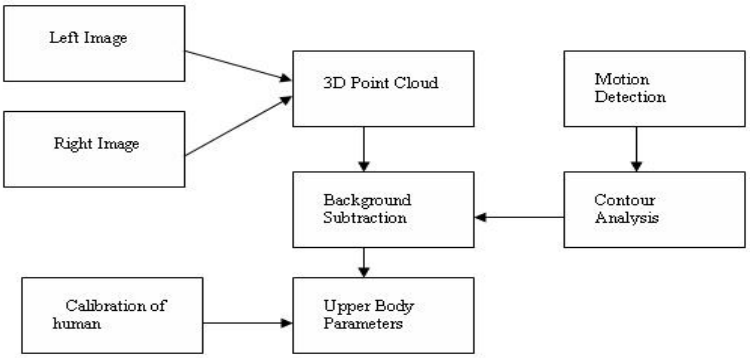

Figure 3: An Overview of our Approach

In this report, the words, video frame and image refer to an image from video at time *t*. Both words have been used interchangeably.

The remainder of this report is organised as follows. Chapter 2 provides the background for human body tracking. The fundamental steps of computer vision namely, video acquisition, pre-processing, segmentation, representation and recognition are discussed in detail with past research. In addition, past approaches to human body tracking and their applicability in public spaces is also detailed. Our approach to tracking human bodies in public spaces (figure 3) is discussed in chapter 3. Results are also presented in this section. Chapter 4 discusses the results, including the limitations of the approach and outlines potential future work. Finally, Chapter 5 derives a conclusion.

[2]www.hitlabnz.org

2 Background

2.1 Human Vision

The basis for learning in humans is the sensory systems of touch, smell, vision, hearing and taste. Out of these, vision and hearing are considered to be a complex process[21]. From the beginning of time humans have tried to explain the complex process of vision. Images and colours are constantly updated as you turn your head and redirect your attention. The seamless quality in the images that we see is possible because human vision updates images, including the details of motion and color, on a time scale so rapid that a break in the action is almost never perceived. The range of color, the perception of seamless motion, the contrast and the quality, along with the minute details, that most people can perceive make real-life images clearer and more detailed than any seen on a television or computer screen. The efficiency and completeness of our eyes and brain is unparalleled in comparison with any piece of hardware ever invented[21].

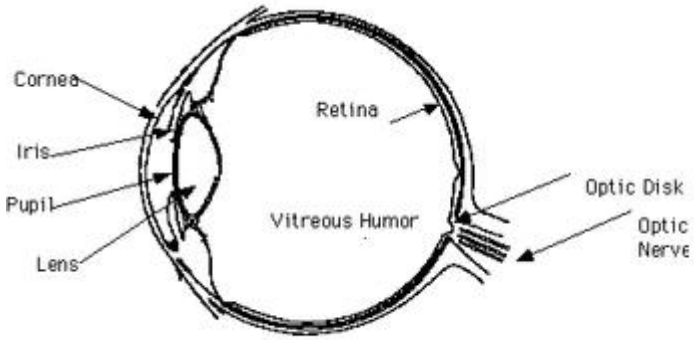

Figure 4: Anatomy of a human eye

Figure 4 shows the anatomical components of the human eye. The main structures are the iris, lens, pupil, cornea, retina, vitreous humor, optic disk and optic nerve. Light that reflects off of objects around us is imaged onto the retina by the lens. The retina, which consists of three layers of neurons (photoreceptor, bipolar and ganglion) is responsible for detecting the light from these images and then causing impulses to be sent to the brain along the optic nerve. The brain decodes these images into information that we know as vision. On the other hand, teaching a computer to see and interpret the real world is very difficult.

2.2 Computer Vision

The analysis of image content and conversion into meaningful descriptions is termed as computer vision. In other words, computer vision is a branch of artificial intelligence and image processing concerned with computer processing of images from the real world. Computer vision research also depends on techniques from a wide range of other fields such as computer graphics and human

computer interaction (HCI). Computer vision uses statistical methods to disentangle data using models constructed with the aid of geometry, physics and learning theory[20].

2.3 Fundamental Steps in Computer Vision

In order to convert images into meaningful descriptions, a computer vision system carries out the following fundamental steps[22] shown in figure 5.

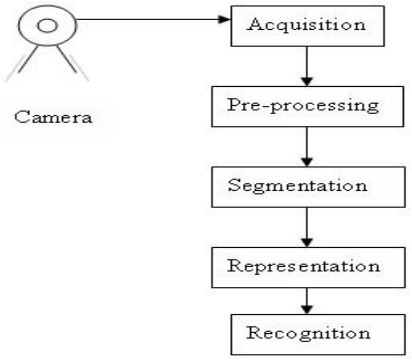

Figure 5: Fundamental Steps in Computer Vision

Each of the above illustrated steps is an active area of research in computer vision. At each stage, input is received from the previous step and output is fed to the next step. Table 1 shows the important research sub-areas at each step.

Step	Sub-area
Acquisition	Monocular vision, Camera calibration, Stereo vision
Pre-processing	Image filtering, Histogram etc.
Segmentation	Background subtraction, Edge detection, Region-oriented segmentation etc.
Representation	Contour extraction, Model-fitting etc.
Recognition	Tracking

Table 1: Research Problems at Each Step

For each of the sub-areas, there has been extensive past and ongoing research. Numerous algorithms have been developed for each of them. For example, there is ongoing research for performing accurate background subtraction. Stereo vision is a well studied problem as well.

It is practically impossible to cover each of the fundamental steps (figure 5) and their related sub-areas in this chapter due to the vast literature. Therefore, this chapter only details the background relevant to developing our body tracking technique (figure 3) for public spaces. The relevant background can be summarized as follows :-

- Stereo vision and camera calibration - We used stereo cameras and calibrated them in our approach to get 3-D information of the real world. Therefore, it is important to understand the principles and past research in this area.

- Image filtering - Removing noise from images acquired from the cameras is also important to get good results.
- Background subtraction - In a public space with varying background, it is desirable to have a robust background subtraction technique.
- Contour extraction - Identifying human shaped contours from edge data and analyzing them has been a popular approach to track humans. Therefore, we discuss various algorithms to extract contours.
- Previous approaches to tracking humans.

2.4 Video Acquisition

The first step in any computer vision application is to input a digital video of the problem domain. This is normally done using digital cameras connected to a computer. Unlike film-based cameras, digital cameras have a image sensor that converts light into electrical charges[23]. The image sensor employed by most digital cameras and web cams is a charge coupled device (CCD). Some low-end cameras use complementary metal oxide semiconductor (CMOS) technology. CMOS technology improves the image quality however it is comparatively slower than CCD cameras. Two important attributes of a camera with respect to computer vision applications are resolution and colour.

The amount of detail that the camera can capture is called the resolution, and it is measured in pixels. The more pixels your camera has, the more detail it can capture. High resolution video frames can enhance the performance of pre-processing and segmentation algorithms in computer vision applications. On the other hand, higher resolution can lead to increased processing power and time. Image sensors use filters to look at the incoming light in its three primary colours red, green and blue. A computer vision system either uses a single camera (monocular vision) or multiple cameras based on the problem domain. Either way, an accurate camera calibration is always a requirement.

2.4.1 Camera Calibration

In any computer vision system, there are discrepancies between observed image features and their actual positions. This discrepancies needs to be minimized. Camera calibration in the context of computer vision is the process of determining the intrinsic and extrinsic parameters of the camera[24]. The intrinsic parameters include the internal geometry of a camera including camera constant, the location of the principal point and corrections for lens distortions. Extrinsic parameters include position and orientation of the camera in an absolute coordinate system from the projections of calibration points in the scene. In many cases, the overall performance of the computer vision system strongly depends on the accuracy of the camera calibration. Camera projection is often modeled with a simple pinhole camera model.

Several methods for camera calibration are presented in the literature. The classic approach [28] that originates from the field of photogrammetry solves the problem by minimizing a nonlinear error function. Due to slowness and computational burden of this technique, closed-form solutions have been also suggested (e.g. [24], [25], [26]). However, these methods are based on certain

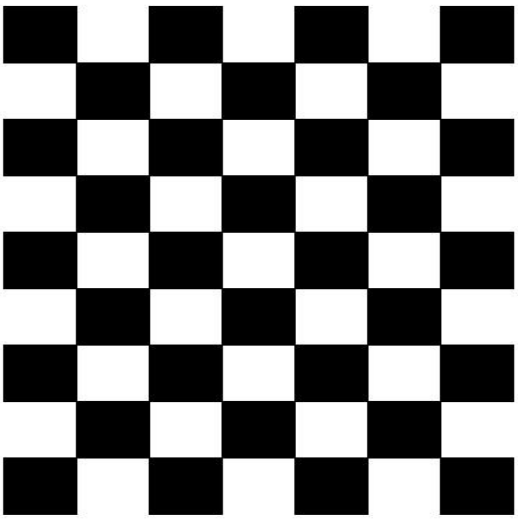

Figure 6: A Calibration Target

simplifications in the camera model, and therefore, they do not provide as good results as nonlinear minimization. There are also calibration procedures where both nonlinear minimization and a closed form solution are used. In these two-step methods, the initial parameter values are computed linearly and the final values are obtained with nonlinear minimization. The methods where the camera model is based on physical parameters, like focal length and principal point, are called explicit methods. In most cases, the values for these parameters are in themselves useless, because only the relationship between 3-D reference coordinates and 2-D image coordinates is required. In implicit camera calibration, the physical parameters are replaced by a set of non-physical implicit parameters that are used to interpolate between some known tie-points (e.g. [27]).

A general strategy for calibration is to view a calibration target such as a checkerboard pattern (figure 6), identify the image points and obtain a matrix called camera matrix using one of the techniques above.

2.4.2 Monocular Vision

In monocular vision, only a single camera is used to obtain a video of the real world. Normally it is used in applications where the depth of the objects in the scene is not required. There are no added geometrical constraints to the camera as is the case with stereo vision. However, camera calibration is essential.

2.4.3 Stereo Vision

Depth information at each pixel can be a useful cue for effiecient background subtraction and 3-D reconstruction of the scene. 3-D information can be estimated indirectly from 2-D intensity images using image cues such as shading and texture[22]. Both shading and texture are considered to be indirect methods and are not accurate. In order to compute real depth at each pixel, stereo vision is used.

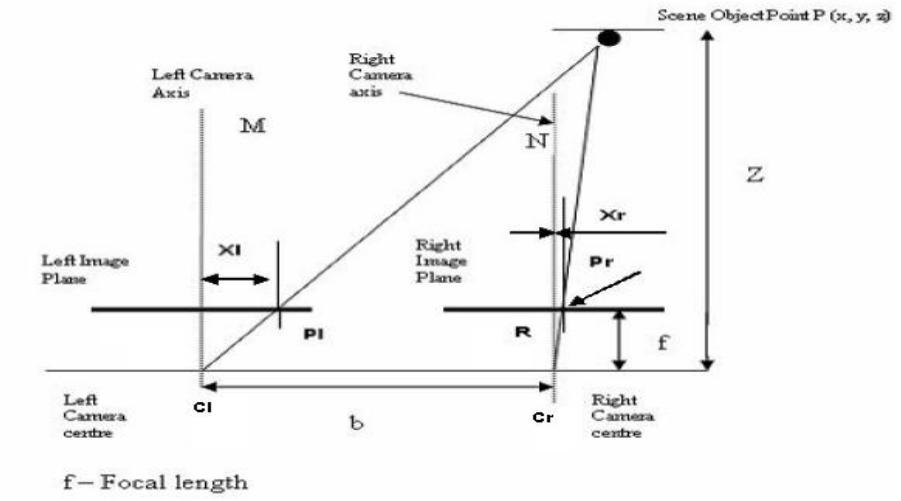

Figure 7: Epipolar Constraint

The epipolar geometry or epipolar constraint for stereo vision is shown in figure 7. In the simplest model, two identical cameras are separated only in x direction by a baseline distance b. In this model, the image planes are coplanar. A feature in the scene is viewed by two cameras at different positions in the image plane. The displacement between the locations of the two features in the image plane is called the disparity. The plane passing through the camera centers and feature point in the scene is called the epipolar plane. The intersection of the epipolar plane with the image plane defines the epipolar line. A conjugate pair is two points in different images that are the projections of the same point in the scene.

In figure 7, the scene point P is observed at points *Pl* and *Pr* in the left and right images, respectively. If we assume that the origin of the coordinate system coincides with the left lens center, then by comparing the similar traingles *P-M-Cl* and *Pl-L-Cl*, we get

$$\frac{x}{z} = \frac{Xl}{f} \tag{1}$$

Similarly from the similar traingles P-N-Cr and Pr-R-Cr, we get

$$\frac{x-b}{z} = \frac{Xr}{f} \tag{2}$$

Combining these two equations, we get

$$z = \frac{bf}{(Xl - Xr)} \tag{3}$$

Thus the depth at various points in the scene may be recovered by knowing the disparities of corresponding image points. Equations for depth values

for cameras in arbitrary position and orientation are more complex and are discussed in [20].

The above mentioned technique is based on the assumption that we can identify conjugate pairs in stereo images. Detecting conjugate pairs has been an extremely challenging research problem know as the correspondence problem. To solve the correspondence problem, for each point in the left image we have to find the corresponding point in the right image. The epipolar constraint significantly limits the search space for finding conjugate pairs.

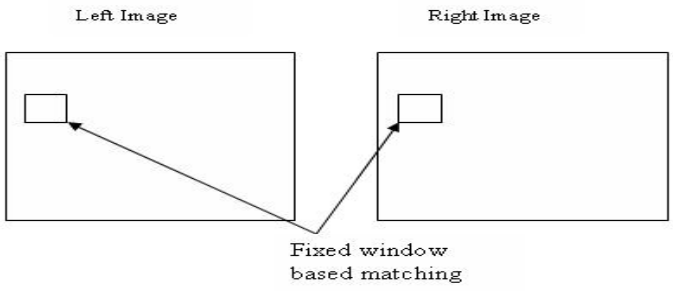

Figure 8: Stereo Matching using Fixed Size Window

Stereo correspondence is an area where extensive research has been done. Stereo algorithms can be classified either as local (window matching as in figure 8) or global. In window-based stereo matching or local matching, the problem is to identify optimal support window for each pixel. An ideal window region should be bigger in textureless regions and should be suspended at depth discontinuties. Fixed window based approaches are invalid at depth discontinuties. Some improved methods, such as adaptive windows[30], shiftable windows[31] and compact windows[32] try to avoid the problems at depth discontinuties. Bayesian methods ([33], [34], [35]) are global methods that try to model discontinuties and occlusion using statistical techniques.

2.4.4 Monocular Vision or Stereo Vision for Public Spaces ?

As discussed above, stereo vision can help determine depth information of the real world. This information can be used for background subtraction and many other segmentation techniques. A lot of ambiguity in the scene could be removed by identifying pixels close to cameras. For example, if a motion capture application is placed in a public space, it would be desirable to identify a human who moves in front of the camera/cameras from a human walking in the background. In addition, disparity values obtained from stereo cameras are less sensitive to lighting. Monocular vision would not fare well as it does not give us any information of the depth of the scene. Although, installing stereo cameras would be expensive, it would serve the purpose. Therefore, in our approach, we decided to use stereo vision with accurate camera calibration techniques. Techniques used to calibrate the cameras and obtain 3-D information are discussed in the next chapter.

2.5 Pre-processing

Whenever an image is acquired by a camera, often the vision system for which it is intended is unable to use it directly. Random variations in intensity, variations in illumination or poor contrast could corrupt the image. This must be dealt with in early stages of vision processing. This is true in a public space as the image capturing device could be exposed to various levels of temperature and change, which could result in noise in the image.

2.5.1 Image Filters

As already mentioned, images can be corrupted by random variations in intensity values, called noise. Some common types of noise are salt and pepper noise, gaussian noise and impulse noise. Salt and pepper noise contains random occurrences of both black and white intensity values. Impulse noise contains only random occurrences of white intensity values. Variations in intensity that are drawn from a gaussian or normal distribution are the source for gaussian noise such as noise due to camera electronics. Figure 9 shows an original image and the same image corrupted with noise[29].

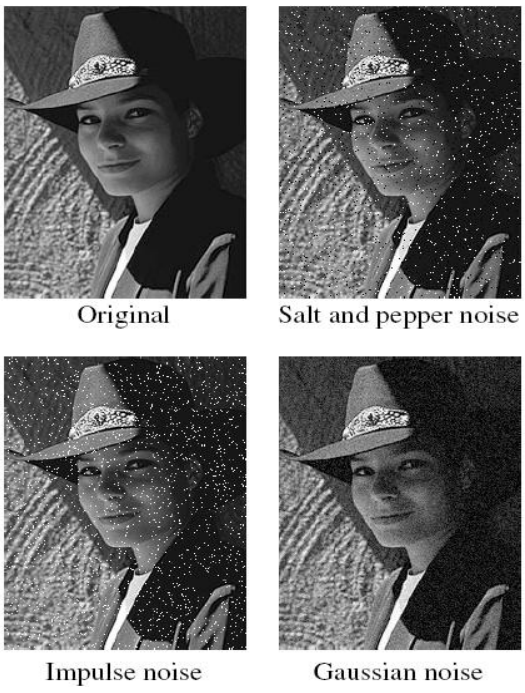

Figure 9: An Image before and after adding Noise

Image filters define a large class of transforms whereby the contents of the images are modified to remove noise. Filters can be classified either in the spatial or frequency domain. In the spatial domain, filters could either be linear or non-linear. Lets look at the spatial domain filters (linear and non-linear) first.

Many computer vision and image processing operations can be modeled as

a linear system. For such a system, when the input to the system is an impulse i(x,y) centered at the origin, the output g(x,y) is the system's impulse response. Furthermore, a system whose response remains the same irrespective of the position of the input is called a space invariant system.

A linear space invariant (LSI) system can be completely described by its impulse response g(x, y) as shown in figure 10. Here f(x, y) and h(x, y) are input and output images, respectively.

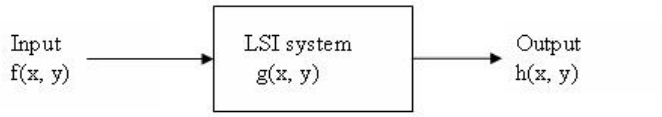

Figure 10: LSI system

The above system must satisfy the relationship:

$$a.f_1(x,y) + b.f_2(x,y) = a.h_1(x,y) + b.h_2(x,y) \tag{4}$$

where $f_1(x,y)$ and $f_2(x,y)$ are input images, $h_1(x,y)$ and $h_2(x,y)$ are the output images corresponding to f_1 and f_2 and a and b are constant scaling factors. The output h(x,y) is the convolution of f(x,y) with the impulse response g(x,y). Convolution is a spatially invariant operation, since the same filter weights are used throughout the image. If f and h are images, convolution becomes the computation of weighted sums of the pixels. The impulse response, g is referred to as convolution mask. The same set of weights are used for different parts of the image. Linear smoothing filters (gaussian filters)are good for removing gaussian noise.

A spatially varying filter requires different filter weights in different parts of the image and hence cannot be represented by convolution. These are regarded as non-linear filters. Two important non-linear filters are median filters and mean filters, where value of each pixel is replaced by the average of all the values in the local neighbourhood[22].

An image can also be represented by its frequency components using a fourier transform. Convolution in image domain corresponds to multiplication in the spatial frequency domain. Therefore, convolution with large filters, which would normally be an expensive process in the image domain, can be implemented efficiently using the fast fourier transform. However, in computer vision, most algorithms are non-linear, so fourier transform methods cannot be used.

2.5.2 Histogram Modification

Sometimes images contain unevenly distributed gray values. Intensity values for such an image lie within a small range. This is true for images with poor contrast. Before performing segmentation or tracking, there is a need to enhance the contrast to get better results. Histogram equalization is a method for stretching the contrast of such images by uniformly redistributing the gray values.

(a) Before (b) After

Figure 11: Enhancing Image Contrast by Histogram Modification

One example of histogram modification is image scaling (figure 11). In this method, the pixels in the range [a, b] are expanded to fill the range $[z_1, z_k]$. The formula for mapping a pixel value z in the original range into a pixel value $z^|$ in the new range is:

$$z^| = \frac{z_k - z_1}{b - a}(z - a) + z_1 \tag{5}$$

$$= \frac{z_k - z_1}{b - a}z + \frac{z_1 b - z_k a}{b - a} \tag{6}$$

Although there are many other complex techniques for histogram modification, they are beyond the scope of this report.

2.5.3 Image Filtering for Public Spaces

Our final goal is to track a human body of interest in a public space. The images received from the stereo camera system, could be subject to noise. Therefore, we applied a few image filters to enhance the quality of images before performing segmentation and tracking. The details are discussed in the next chapter.

2.6 Segmentation

Segmentation refers to identifying groups or regions of connected pixels with similar properties. These regions are important for the interpretation of an image because they may correspond to objects in a scene. An image could contain several objects and each object may have several regions corresponding to different parts of the object. For an image to be interpreted accurately, it must be partitioned into regions that correspond to objects or parts of an object.

2.6.1 Background Subtraction

Background subtraction is a technique for identifying moving objects in image sequences. In other words, its purpose is to identify and discard background pixels in image sequences. The traditional approach for background subtraction assumes a stationery background model. This will not work well for backgrounds

that are not stationary, which is true in public spaces. Lets look at the various background subtraction techniques in detail.

The basic background subtraction technique uses a frame difference of the current frame with the stationery background frame to identify a foreground object. In other words, every pixel in the current frame is compared to the pixel at the same position (RGB color values or gray scale values) in the stored background image. If they are similar, then the pixel is a part of background and vice versa. A threshold is applied to account for intensity changes. It can be illustrated as follows:

$$|frame_i - background_i| > Threshold \tag{7}$$

This technique is very sensitive to threshold and is not adaptable to changing background. A slightly improved approach is to perform frame difference on two consecutive frames as follows:

$$|frame_i - frame_{i-1}| > Threshold \tag{8}$$

This is again sensitive to threshold and cannot account for background changes, illumination changes etc. Histogram methods were proposed whereby the background model at each pixel location is based on pixel's recent history. One such method called average or median calculates the value of a pixel based on average of previous n frames[36]. Background can also be removed if stereo vision is used and depth value at each pixel is obtained.

More complex methods are based on fitting one gaussian distribution (μ, σ) over the image histogram[17]. This gives us the background probability density function (PDF). The background PDF is updated using a running average as follows:

$$\mu_{i+1} = \alpha F_i + (1 - \alpha)\mu_i \tag{9}$$

$$\sigma^2_{i+1} = \alpha(F_i - \mu_i)^2 + (1 - \alpha)\sigma^2_i \tag{10}$$

where F_i is the frame i. Another complex technique called mean-shift based estimation[37], uses a gradient-ascent method to be able to detect the modes of a multimodal distribution together with their covariance matrix. Table 2 summarizes the important background subtraction techniques.

Method	Advantages	Disadvantages
Static background	Simple, fast	Not adaptable to background changes
Frame difference	Simple	Sensitive to threshold
Running average	fast	Memory consuming
Depth	Accurate depth at each pixel	Two cameras (minimum), processing time
Gaussian average	reliable	Cannot cope with multimodal backgrounds
Mean-shift based	reliable	Memory consuming

Table 2: Background Subtraction Techniques

2.6.2 Background Subtraction in a Public Space

In our approach, we use the 3-D information from the cameras along with the extracted human shaped contours to determine the depth of each pixel inside the contour. This can help to exclude the other pixels in an image. Frame difference and running average techniques are sensitive to threshold and result in lot of noise. However, the accuracy of our background subtraction would depend on detecting human shaped contours in the images. In order, to determine contours, first, edges need to be detected.

2.6.3 Edge Detection

Background subtraction produces a segmentation that yields all the pixels that, in principle, belong to the object or objects of interest in an image. An alternative to this is to find those pixels that belong to the borders of the objects or edges. Edges occur at boundary between two different regions in an image where there is a significant change. A variety of techniques are available which determine the magnitude of contrast changes and their orientation.

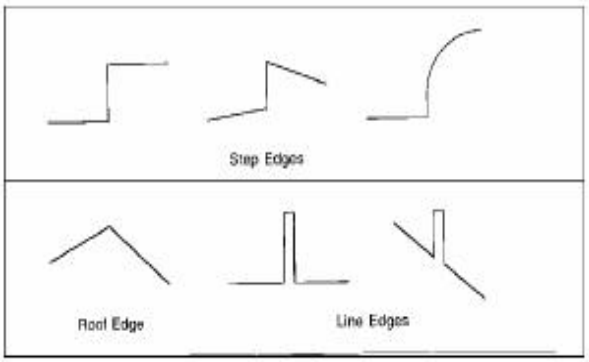

Figure 12: Types of Edges

An edge in an image is a significant local change in the image intensity. Discontinuities in the image intensities can either be *step* discontinuities, where the image intensity abruptly changes from one value on one side of the discontinuity to a different value on the opposite side or *line* discontinuities, where the image intensity abruptly changes value but then returns to the starting point with some short distance. Because of smoothing produced in some devices and filtering, line edges might become roof edges.

An edge detector is an algorithm that produces a set of edges (edge points) from an image. The coordinates of an edge point may be with respect to image coordinates. There are three steps to edge detection namely filtering, enhancement and detection[22].

Taking an edge to be a change in intensity taking place over a number of pixels, edge detection algorithms generally calculate a derivative of this intensity change. Let's consider the detection of an edge in 1 dimension (figure 13). In this instance, our data can be a single line of pixel intensities. For instance an edge can clearly be detected between the 4th and 5th pixels in the following 1-dimensional data:

5	7	6	4	145

Figure 13: One Dimensional Edge Data

Many edge-detection operators are based upon the 1st derivative of the intensity which gives us the intensity gradient of the original data. Using this information we can search an image for peaks in the intensity gradient. If I(x) represents the intensity of pixel x, and I'(x) represents the first derivative (intensity gradient) at pixel x, we therefore find that:

$$I'(x) = -1.I(x-1) + 0.I(x) + 1.I(x+1) \tag{11}$$

Some other edge-detection operators are based upon the 2nd derivative of the intensity. This is essentially the rate of change in intensity gradient and is best at detecting lines. To find lines, we can search the results for zero-crossings of the change in gradient. If I(x) represents the intensity at point x, and I"(x) is the second derivative at point x:

$$I"(x) = 1.I(x-1) - 2.I(x) + 1.I(x+1) \tag{12}$$

Some important first-order edge detection operators are Prewitt, Roberts Cross, Sobel and Canny. Laplacian is an important second order operator. These operators are decades old and are discussed in a number of computer vision and image processing texts such as [22], [38] and [20]. Figure 14 shows an edge detection result by using a Sobel operator.

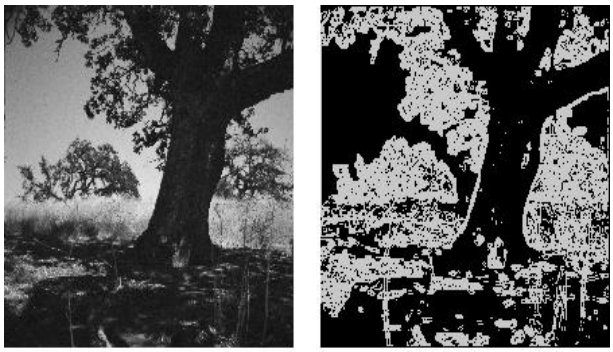

Figure 14: Sobel Edge Detection

2.7 Representation

2.7.1 Contour Extraction

Edge detectors typically produce short, disjoint edges. These edges must be linked together into a representation for a region boundary. This representation is called a contour. In body tracking, we need a region which outlines the

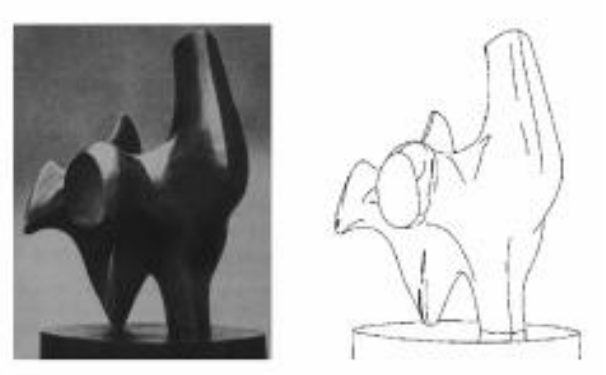

Figure 15: An Example of Contour Extraction

human body. There are two main categories of methods for contour extraction as follows:-

- Local Methods - These methods look in the local neighbourhood to extend edges.
- Global Methods - Domain knowledge is incorporated to achieve contour extraction. It is computationally more expensive.

Figure 15 shows an example of a contour extracted from an image. Contour extraction is normally done in terms of chain codes. Chain codes are a notation for recording the list of edge points along a contour. Two important chain code approximations are Freeman chain code[39] and Teh-Chin chain approximation[40]. However, the Freeman code approximation suffers from quantization errors, missed points or redundant points[41]. In our approach, we used Teh-Chin chain approximation code which is discussed in chapter 3.

2.8 Tracking

2.8.1 Tracking Techniques

Tracking refers to identifying feature points or interesting points in a video frame and predicting its position in the next frame. After extracting the contour of the human body, it is desirable to keep track of it in the each frame as the human body moves around. There are many tracking techniques available. One can use the configuration in the current video frame and a dynamic model to predict the next configuration as in [4],[42]. Another common approach is using kalman or particle filters[43]. Particle filtering uses multiple predictions which are obtained by running samples of prior through a model of dynamics. There could be problems, for example, if an arm swings past an "arm-like" pole, the correct local maximum must be found to prevent the track from drifting. Annealing the particle filter is one way to attack the difficulty, which is discussed in [44]. Some researchers strongly model the dynamics in order to track. Another alternative is to ignore dynamics and track objects of interests in each frame independently using cues such as local motion or appearance[45].

2.9 Human Body Segmentation and Tracking

The literature on tracking a human body is quite vast. In [17], color-based segmentation is used to track people with a single camera. This is not helpful in our case as public spaces are unconstrained and therefore color in the scene could vary. Cardboard models[5] were proposed to track articulated motion of human limbs represented by a set of connected planar patches. Figure 16 shows how human bodies were modeled using rectangular planes. Bregler and

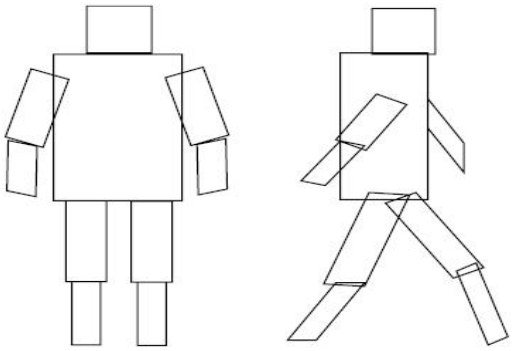

Figure 16: Cardboard Models

Malik[4] track joint angles of a 3D model of a human body using optical flow measurements from a video sequence. Their results are summarized in figure 17. Although, there results are quite good, they use controlled lighting and clothing to achieve this. In a public space, this would not work well.

Figure 17: Optical Flow Techniques to Track Humans

Gavrilla and Davis[11] is another example of a model-based technique. In [3], mikic et al. acquire human body model and track the motion using input from multiple synchronized video streams. The video frames are segmented and the 3-D voxel reconstructions of the human body shape in each frame are computed from foreground silhouettes. Jojic et al.[6], track articulated structures in dense disparity maps obtained from stereo image sequences. Ramanan and Forysth[2], build up a model of appearance of a body of each individual by clustering

candidate body segments and then use this model to find individuals in each frame. Mori and Malik[1], store a number of different exemplar 2-D views of the human body in a variety of different configurations and viewpoints with respect to the camera. The joint positions are labeled on each view. The shape obtained from each frame is then matched against the views to find the joint positions. This is not applicable in our case since we deal with 3-D data. Most of the above mentioned techniques work well in a constrained environment. They have not been tested in a public space. Kalman and particle filters work well for an object whose motion is very predictable. An articulated object such as a human body can move around very freely that the track can be lost. More robust approaches are needed to deal with public space complexities.

3 Our Approach

3.1 Aims

The previous chapter outlined the past research on human body tracking. Most methods for body tracking work well in laboratory conditions. Many researchers have used skin color to segment and track various parts of human bodies. In a public space, the clothing of a person could vary. This would also make it impossible to rely on any skin color as parts of body could be covered. Some results from previous research do not rely on skin color, however, they use a static background model. Once again, in a public space background will not remain static. Apart from this, lighting and reflectance also varies in the scene. Therefore, the main aim of this research is to develop a technique for human body tracking in such an unconstrained environment. In particular, we try to address the following two questions:-

- How do we robustly track humans in a public space ?
- How do we retrieve joint parameters of the upper human body ?

The upper body parameters we have decided to retrieve for this research are as follows:-

- Center of mass of a human body
- Principal axis of a human body
- Torso position
- Shoulder positions
- Head center of mass

3.2 Method

3.2.1 Research Scenario

We address the research questions (section 3.1) by considering a motion capture application working in a public space. Motion capture is termed as recording of human body movements for immediate or delayed playback by a graphic character. Motion capture is one of the many applications which could benefit from markerless computer vision tracking in a public space. Although there are many other applications such as surveillance and gesture-based interface, we selected motion capture because of our earlier involvement in developing[3] such an application called LOTR (*Lord of the Rings*). However, this application was not robust enough for use in a public space. Therefore, in this research, we propose to make such applications robust enough for public spaces. Before we go into details of our approach, lets discuss the LOTR as a simple motion capture application and its limitations.

[3]Feb-May 2004

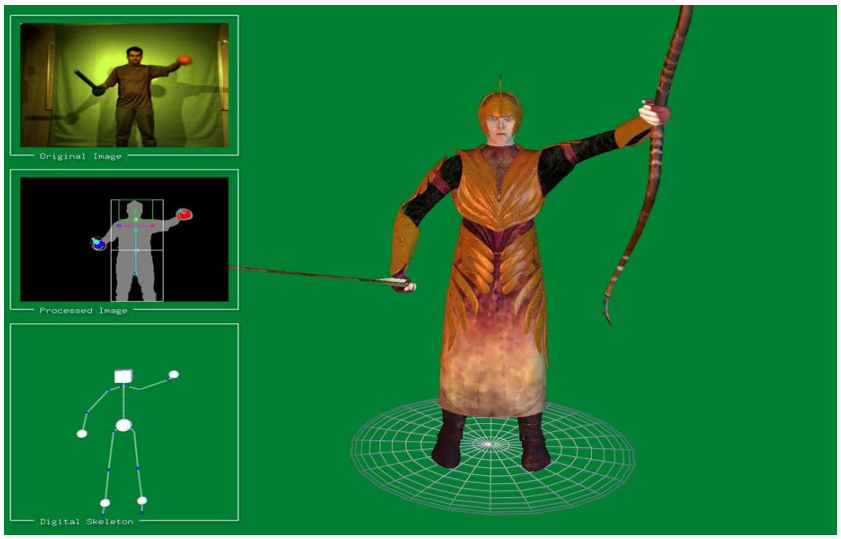

Figure 18: Lord of the Rings

3.2.2 LOTR and its Limitations

LOTR is a simple motion capture tool designed at the HITLab earlier this year. It contains a front projected display with graphical characters rendered on it (refer figure 18). Characters rendered are one of *uruk*, *gondorian* or *elf*. Stereo cameras are attached below the display. A person walks in front of the display with a yellow colored sword in one of his hands and a red shield in other. As the person performs body movements, the graphical character on screen replicates it. The figure shows a graphic character called *uruk* and three small windows on the left which illustrate the computer vision processing done on each video frame to identify body positions. The topmost image shows the actual input from the cameras. The middle image shows a background subtracted image (blob) along with some identified body positions using coloured lines. There are some limitations of this application. A green background was used in this application to assist in background subtraction. In other words, a static background model was used. In addition, known colors for sword and shield were used to help track arm positions in each frame (colour tracking). Using a known colour sword and shield would also classify this application as a marker-based computer vision technique.

In our approach, we do not rely on any colour tracking. In addition, we also consider people moving behind the scenes, which is true in a public space.

3.2.3 Hypotheses and Research Setup

We consider a motion capture application similar to LOTR in a public space with stereo cameras attached to it. This setup is shown in figure 19. The display would show a graphic character that would replicate motion of a human who walks in front of the stereo cameras and then, moves his body parts for a few seconds (could vary as per interest). Then he walks out from the front

of the display and someone else may enter in. The background in the scene is changing as is the lighting and reflectance. The projector and stereo cameras are connected to a workstation where processing of live video is carried out and graphic characters on screen are updated. We installed this setup at the HITlab.

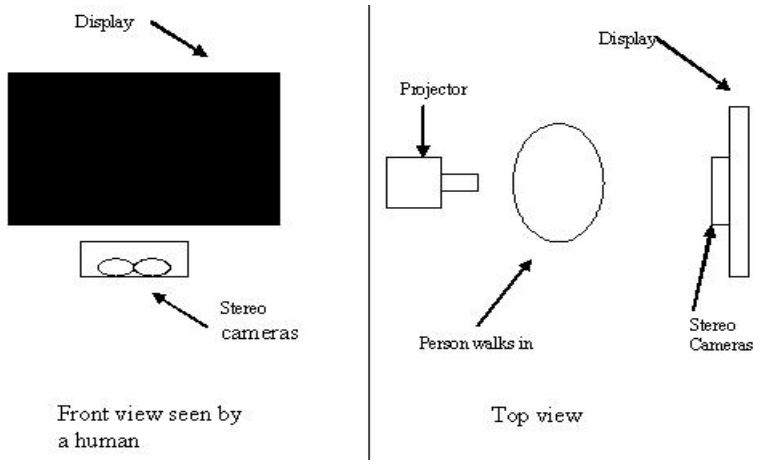

Figure 19: Research Setup

We assume that a person walks in front of the cameras, calibrates for a few frames and then moves his body parts. Calibration of a person refers to the subject standing straight in front of the camera with hands not close to the body. This is useful because some estimates of the human body such as height and width can be approximated using this step. We also assume that no more than one person can interact with the display at any given time.

3.2.4 Hardware and Software Requirements

The hardware and software for this research has been provided by the HITlab. They can be summarized as follows:-

Software

- Open source Computer Vision library (OpenCV) version beta 3 - This library contains numerous implementations of computer vision and image processing algorithms
- Microsoft Visual Studio 6 Integrated Development Environment (IDE)

Hardware

- SRI Small Vision Systems (SVS) variable baseline stereo cameras[4] - This is a commercially available stereo camera system. We decided to use this because of real-time requirements. The details of this are explained in section 3.2.5
- 2.8 GHz Intel workstation
- Video projector

[4] www.videredesign.com

3.2.5 Stereo Cameras and Calibration

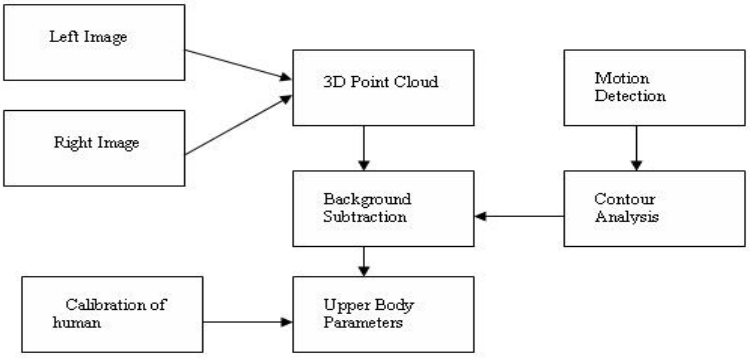

Figure 20: A Summary of our Approach

Our approach is illustrated in figure 20. The important step was to decide how to obtain 3-D point cloud from the real world. The next two paragraphs explain our initial approach and how the SVS stereo system was more suitable.

Initial Approach

We used two *Logitech Quick Cam Pro 4000* CCD cameras inorder to obtain 3-D point cloud. OpenCV library was used for accessing video frames from the cameras. Each of the cameras was calibrated using a checkerboard pattern (figure 6) and OpenCV functions, which calculate the camera parameters. Then disparity values were returned by OpenCV functions (using [30]) and we used equations 1, 2 and 3 to calculate the depth value for each pixel. The 3-D points returned were prone to noise. This is because the cameras were not mounted on a fixed rig and a minor change in camera position would increase noise. In addition, the frame rate dropped down to 3 frames per second due to high processing power required by the stereo algorithms. Because faster result was desirable, we decided to use commercially available software.

SVS Stereo Cameras

Figure 21: SVS Camera System

SVS stereo system is a commercially available setup by Videre Design[5]. Although, there are many different variations of the stereo head, we selected the variable baseline model *STH-MDCS-VAR-C*. It comes with two camera lenses mounted on a fixed rig (figure 21). The baseline distance between the two cameras can be varied. It also includes an applications programmer interface (API). There are functions to compute disparities, 3-D points and frame captures.

The first step was to calibrate the cameras using checkerboard patterns which is illustrated in Appendix A. The video frame returned from this system was of a different format from OpenCV image format. As OpenCV has many high-level computer vision functions implemented inside it, we decided to convert a SVS image to OpenCV image format. Since the camera system has an onboard processor, the frame rate was high at 30 frames per second to achieve real-time stereo. The resolution of a video frame was 320 x 240 pixels.

3.3 Estimating Body Positions and Results

In this research, we ignore dynamics and find the human body and its parameters in each frame using cues such as local motion and depth. The whole approach is explained in this section in sequential order.

3.3.1 3D Point Cloud

Figure 22: Result of a 3-D Point Cloud

As already discussed, we used the SVS camera setup and OpenCV library to obtain a 3-D point cloud. A snapshot of 3-D points obtained is shown in figure 22. The right image shows the disparity video frame in which level of green is used to distinguish depth of a pixel. That is, high value of green is classified as close to camera. Black colour is used to denote pixels very far from the camera. As seen in the result, some regions of the image, although in front of the camera are classified as black (far from cameras). This is because of lighting and reflectance variation in the room.

3.3.2 Detection of Motion in the Scene

It is important to detect motion in front of the system in a public space. In our approach, we first detect if there is any motion in the scene. If there is

[5]www.videredesign.com

motion, then we carry on to the remaining processing steps. An ideal technique to detect motion in a video with static background is to compare the pixel's current intensity value with the intensity value of the same pixel in the static background frame as follows:

$$|frame_i - background_i| > Th \tag{13}$$

Th is the threshold which is set by observing the best result.

Since we have a changing background, we first convert each frame to grayscale format and use difference between two consecutive frames with a certain threshold. This is as follows:

$$|frame_i - frame_{i-1}| > Th \tag{14}$$

3.3.3 Finding Contours

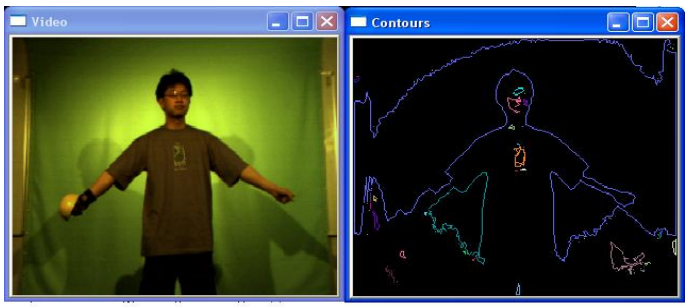

Figure 23: Result of Contour Detection

Contours can be approximated to find regions where pixels could belong to an object. We used OpenCV library for this project, since it has many advanced image processing functions implemented. Teh-Chin chain approximation algorithm[40], was used to find contours in the video frames. It returns various contours including an approximate contour of a human being, if in front of the cameras. This is shown in figure 23. Edges for each contour are drawn in different colours. Some other small contours illustrated with different colours were also found by the Teh-Chin method. However, we find the area of each contour in pixels and neglect contours with smaller areas.

3.3.4 Background Subtraction

The problem of background subtraction is much simplified once we have accurate information from the above mentioned steps. We can find the depth of each pixel inside the human shaped contour. This process can help detect a contour (human) close to the camera. Motion close to the camera would be regarded as of interest. In addition, once a person calibrates in front of the cameras and goes away, we build a model or a 'sweet spot' in front of the camera. If we again identify motion in the sweet spot, we would classify that as motion of interest.

3.3.5 Finding Joint Positions of Upper Human Body

Once we identify a human in the foreground, the next step is to find the center of mass, principal axis, head and shoulder positions in each frame.

Center of Mass

We calculate the first order moments to find the center of mass of the person as follows:

$$\bar{x}\sum_{i=1}^{n}\sum_{j=1}^{m}B[i,j] = \sum_{i=1}^{n}\sum_{j=1}^{m}jB[i,j] \tag{15}$$

$$\bar{y}\sum_{i=1}^{n}\sum_{j=1}^{m}B[i,j] = \sum_{i=1}^{n}\sum_{j=1}^{m}iB[i,j] \tag{16}$$

where $\bar{x}$ and $\bar{y}$ are the coordinates of the center of the region.

Principal Axis

The center of mass along with principal axis can be used to estimate the angle of the upper body. This can be then used to update the graphic model. Knowing the center of mass, we can use second order moments to find the principal axis as follows. We find $x^{|}$ and $y^{|}$ for each point (x,y) on the body using the values of $\bar{x}$ and $\bar{y}$ as:

$$x^{|} = x - \bar{x} \tag{17}$$

$$y^{|} = y - \bar{y} \tag{18}$$

Then we solve for a, b and c as follows:

$$a = \sum_{i=1}^{n}\sum_{j=1}^{m}(x_{ij}^{|})^2B[i,j] \tag{19}$$

$$b = 2\sum_{i=1}^{n}\sum_{j=1}^{m}(x_{ij}^{|})(y_{ij}^{|})B[i,j] \tag{20}$$

$$c = \sum_{i=1}^{n}\sum_{j=1}^{m}(y_{ij}^{|})B[i,j] \tag{21}$$

The orientation of the principal axis is then given by:

$$sin2\Theta = \pm\frac{b}{\sqrt{(b^2+(a-c)^2)}} \tag{22}$$

$$cos2\Theta = \pm\frac{a-c}{\sqrt{(b^2+(a-c)^2)}} \tag{23}$$

Torso Positions

Pixels above the center of mass are then classified are belonging to the upper human body. Again moments are calculated using those pixels to find the torso position. The same can be done on the lower human body part.

Head Center of Mass

When a person calibrates in front of the cameras, we estimate the top of the head in the background subtracted image. We then build a small rectangular region around the face and find moments of the pixels in that region to find the center of mass of the face. The principal axis of the face can also be estimated in the same way.

Shoulder Positions

It is generally observed that the shoulders are 90 degrees apart from the torso for a normal person. Therefore, when a person calibrates in the initial stage, the estimates of the shoulder can be found starting from the center of mass. We search left in the image until we find a black pixel (background) and then search upwards until we find a white pixel (foreground). This search operation helps to find the approximate width of a person. Once we get an approximate width of the shoulders from the above process, we use that distance and estimate the shoulders as 90 degrees from the torso on each side.

3.4 Integrated System Results

We decided to test our approach on sample videos of a person walking in front of the display and performing body movements. While one person is doing that, another person walks behind the scenes to give an effect of a public space. The lighting conditions in the room were made very unstable by having different lights on along with the natural sunlight entering the room. The person had no restriction on any kind of clothing to wear. We gathered around 200 frames of video and applied our technique to identify his upper body positions. Results are shown in figure 24, 25 and 26.

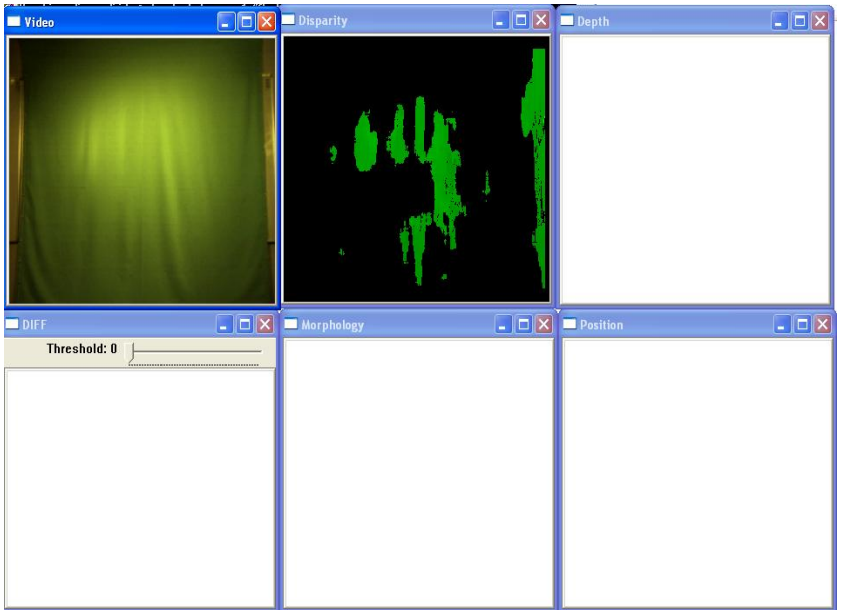

Figure 24: Frame A of Result

Figure 24 shows the view as seen from the cameras when no person is in the scene. There are six windows in these results. The top left image shows the video, the top middle shows the disparity image obtained from the camera and the top right shows the result of contours processing and background subtraction. The image at the bottom left shows the results after motion detection in the scene. The image of the bottom right shows results of joint positions detection.

Figure 25: Frame B of Result

Blue line indicates the principle axis. Red circles are used to denote center of mass, torso and shoulder positions. These positions are updated per frame. In figure 25, the person moves his arm and his body positions are tracked which is shown on the bottom right window.

Figure 26: Frame C of Result

In figure 26, a person walks behind the scenes, and once again we identify the human close to the camera and track his body positions.

Results of Body Positions on Sequential Frames

Body positions returned by three frames, frame 5 (figure 27), frame 10 (figure 28) and frame 15 (figure 29) are shown. It can be seen that the principal axis (blue line) and other body positions remain quite robust.

Figure 27: Frame 5

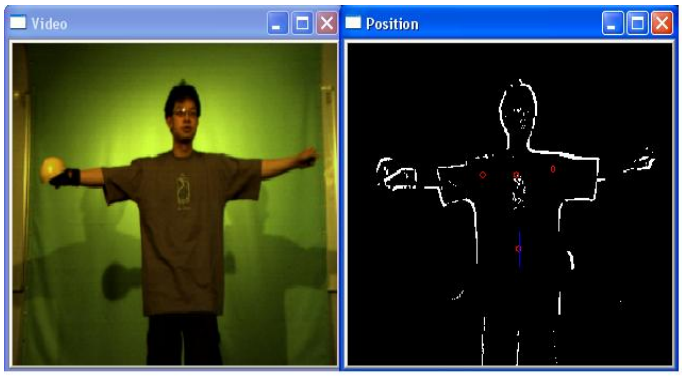

Figure 28: Frame 10

Figure 29: Frame 15

Achievement of Real-time

Table 3 shows the various parts of our approach and how they affected the goal to achieve real-time. The integrated system results discussed above run at a frame rate of 20 frames per second.

Part	Frame Rate
3-D Point Cloud	30 (Original)
After Motion Detection	28
After Contour Processing	23
After Background Subtraction	21
After Estimation of Parameters	20

Table 3: Frame Rate after each Part

4 Discussion and Future Work

4.1 Discussion and Limitations of the Approach

Disparity results obtained from stereo cameras are robust enough as discussed in the last chapter. Accurate camera calibration would always be a important factor. The accuracy of camera calibration would depend on number of factors such as technical knowledge of the person who calibrates it and temperature of the camera system. Contour detection using Teh-Chin algorithm (section 3.3.3), retrieves the contour of a human in front of the cameras. Some other minor contours are also detected. However, such noise has been removed from the system at the cost of processing power. Contour detection consumes a lot of processing as can been seen with the results of drop in frame rate to 20 frames per second. It is important to optimize the approach by having a frame rate of 25 frames per second, which is considered to be real-time. The body positions returned from the integrated system as shown in figures 27, 28 and 29 can be fed into a motion capture application. When a person moves behind the scenes as is the case with figure 26, the principal axis of a human seems to drift away from the expected position. This is due to limitations of contour detection, which considers the person walking behind and the person standing in front of the cameras inside the same contour. Apart from that, the other results indicate that the joined positions marked on each frame by blue lines and red circles are sufficiently robust. The research does not explain how occlusions would be handled, which could be a vital matter in a public space, especially for surveillance applications.

4.2 Future Work

There is lot of potential work for extending this research. The same approach can be extended to track the lower body parameters such as knee and thigh positions. This would make it a full body tracking application. In this research, we assumed that only one person interacts with the motion capture display at any given time. However, there is no reason why more than one person should not be able to interact with the system. Another interesting future work could be using techniques such as reverse registration to identify elbow positions. Occlusions are difficult to track in a public space. There could be some potential work to develop a technique to handle occlusions along with our approach on body tracking. In this research, we considered a motion capture application in a public space. There is potential work to test the same approach for other applications such as vision-based gestural interface in a public space.

5 Conclusion

This research presented a novel approach for tracking human body in a public space. In particular, retrieval of human body positions using stereo cameras and other computer vision techniques was also discussed. The approach was based on markerless vision-based tracking. This is vital for applications such as gesture interface, surveillance and motion capture which have a potential to operate in a public space. Although this report focused on a motion capture application in a public space, other applications mentioned above could also benefit from this research. The report also investigated previous research on human body tracking, which has been of great interest to the computer vision community. The fundamental steps for any computer vision system and their applicable sub-areas for a public space have also been discussed. Various important algorithms at each step are also explained. The results presented in this research are sufficiently robust to varying background, lighting, colour, clothing and reflectance with some limitations. There is lot of potential for research on tracking human bodies which could drive the next generation of multimodal interfaces.

References

[1] G. Mori and J. Malik. Estimating Human Body Configurations using Shape Context Matching. In European Conference on Computer Vision, 2002.

[2] D. Ramanan and D. Forsyth. Finding and Tracking People from the Bottom Up. In IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Volume: 2 , Pages: II-467-II-474, 18-20 June 2003.

[3] I. Mikic, M. Trivedi, E. Hunter and P. Cosman. Human Body Model Acquisition and Tracking Using Voxel Data. International Journal of Computer Vision 53(3), Pages: 199-223, 2003.

[4] C. Bregler and J. Malik. Tracking People with Twists and Exponential Maps. In IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Pages: 8-15, 23-25 June 1998.

[5] S. Ju, M. Black and Y. Yacoob. Cardboard People: A Parameterized Model of Articulated Image Motion. In Proceedings of Second International Conference on Automatic Face and Gesture Recognition, Pages: 38-44, 1996.

[6] N. Jojic, M. Turk and T.Huang. Tracking Self-Occluding Articulated Objects in Dense Disparity Maps. In Proceedings of International Conference on Computer Vision, Corfu, Greece, 1999.

[7] C. Colombo, A. Del Bimbo and A. Valli. Real-Time Tracking and Reproduction of 3D Human Body Motion. In Proceedings of 11th International Conference on Image Analysis and Processing. Pages: 108-112, 26-28th September, 2001.

[8] N. Jojic, B. Brumitt, B. Meyers, S. Harris and T. Huang. Detection and Estimation of Pointing Gestures in Dense Disparity Maps. In Fourth IEEE International Conference on Automatic Face and Gesture Recognition. Pages: 468-475, 28-30th March 2000.

[9] Y. Huang and T.S. Huang. Model-Based Human Body Tracking. In 16th International Conference on Pattern Recognition. Volume: 1, Pages: 552-555, 11-15th August 2002.

[10] N. Howe and A. Deschamps. Better Foreground Segmentation Through Graph Cuts. http://www.arxiv.org/abs/cs.CV/0401017.

[11] D. Gavrilla and L. Davis. 3D Model-Based Tracking of Humans in Action: A Multi-View Approach. In Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Pages: 73-80, 1996.

[12] D.M. Gavrilla. The Visual Analysis of Human Movement: A Survey. Computer Vision and Image Understanding, 73(1), Pages: 82-98, 1999.

[13] H. Ning, L. Wang, W. Hu and T. Tan. Model-Based Tracking of Human Walking in Monocular Image Sequences. In Proceedings of IEEE Region 10 Conference on Computers, Communications, Control and Power Engineering. Volume: 1, Pages: 537-540, 28-31st October 2002.

[14] M. Isard and A. Blake. Contour Tracking by Stochastic Propagation of Conditional Density. In Proceedings of European Conference on Computer Vision. Volume: 1, Pages: 343-356, 1996.

[15] J. Shi and C. Tomasi. Good Features to Track. In Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Pages: 593-600, 1994.

[16] H. Yoshimoto, N. Date and S. Yonemoto. Vision-based real-time Motion Capture system using Multiple Cameras. In Proceedings of IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems. Pages: 247-251, 30th July - 1st August 2003.

[17] C. Wren, A. Azerbayejani, T. Darrell and A. Pentland. Pfinder: Real-Time Tracking of the Human Body. In IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI). 1997

[18] K. Konolige. Small Vision Systems: Hardware and Implementation. Eighth International Symposium on Robotics Research. Japan, October 1997.

[19] Articulated Object. http://www.olympus.net/personal/mortenson/preview/definitionsa/articulatedobject.html.

[20] D. Forysth and J. Ponce. Computer Vision, A Modern Approach, 2003.

[21] D. Szaflarski. How We See: The First Steps of Human Vision. http://www.accessexcellence.org/AE/AEC/CC/vision_background.html.

[22] R. Jain, R. Kasturi and B. Schunck. Machine Vision, Pages: 112-138, 1995.

[23] How Digital Cameras Work. http://electronics.howstuffworks.com/digital-camera.htm

[24] R. Tsai. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off- shelf TV cameras and lenses. In IEEE Journal of Robotics and Automation RA-3(4), Pages: 323-344, 1987.

[25] Y. I. Abdel-Aziz and H. M. Karara. Direct linear transformation into object space coordinates in close-range photogrammetry. Proceedings and Symposium on Close-Range Photogrammetry, Urbana, Illinois, pages: 1-18, 1971.

[26] J. Weng, P. Cohen and M. Herniou. Camera calibration with distortion models and accuracy evaluation. In IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-14(IO), Pages: 965-980, 1992.

[27] G. Q. Wei and S . D. Ma. A complete two-plane camera calibration method and experimental comparisons. Proceedings of 4th International Conference on Computer Vision, Berlin, Germany, pages: 439-446, 1993.

[28] Slama, C. C. Slama. Manual of Photogrammetry, 4th ed., American Society of Photogrammetry, Falls Church, Virginia, 1980.

[29] Linear Image Transforms. http://www-2.cs.cmu.edu/afs/andrew/scs/cs/15-463/pub/www/Lectures/filtering1.pdf

[30] T. Kanade and M. Okutomi. A Stereo Matching Algorithm with an Adaptive Window: Theory and Experiment. Pattern Analysis and Machine Intelligence (PAMI), 16(9), Pages: 259-268, 1994

[31] A.F. Bobick and S.S. Intelli. Large Occlusion Stereo. International Journal of Computer Vision (IJCV), 33(3), Pages: 1-20, 1999.

[32] O. Veksler. Stereo Matching by Compact Windows via Minimum Ratio Cycle. ICCV, 2001.

[33] P.N. Belhumeur. A Bayesian-approach to Binocular Stereopsis. International Journal of Computer Vision (IJCV), 19(3), Pages: 237-260, 1996.

[34] D. Geiger, B. Ladendorf and A. Yuille. Occlusions and Binocular Stereo. International Journal of Computer Vision (IJCV), 14(3), Pages: 211-226, 1995

[35] H. Ishikawa and D. Geiger. Occlusions, Discontinuities and Epipolar Lines in Stereo. ECCV, 1998.

[36] L.M. Fuentes and S.A. Velastin. People tracking in surveillance applications. In 2nd IEEE International Workshop on Performance Evaluation on Tracking and Surveillance, PETS 2001, Kauai (Hawaii-USA), 2001.

[37] B. Han, D. Comaniciu and L. Davis. Sequential Kernel Density Approximation through Mode Propagation: Applications to Background Modeling. Asian Conf. Computer Vision (ACCV'04), Jeju Island, Korea, 2004.

[38] R. Gonzalez and R. Woods. Digital Image Processing, Pages: 429-435, 1993.

[39] H. Freeman and L. S. Davis. A corner-finding algorithm for chaincoded curves. In IEEE Computer Transactions., vol: C-26, pages: 297-303, March, 1977.

[40] C. H. Teh and R. H. Chin. On the Detection of Dominant Points on Digital Curves. In IEEE Transactions on Pattern Analysis and Machine Intelligence. Volume: 11, August, 1989.

[41] J.A. Horst. Efficient Piecewise Linear Approximation of Space Curves using Chord and Arc Length. In Proceedings of the SME Applied Machine Vision 96 Conference, Cincinnati Ohio, June 3-6, 1996

[42] K. Rohr. Incremental Recognition of Pedestrians from Image Sequences. In IEEE Conference on Computer Vision and Pattern Recognition. Pages: 9-13, 1993.

[43] H. Sidenbladh, M.J. Black and D.J. Fleet. Stochastic Tracking of 3-D Human Figures from 2-D Image Motion. In European Conference on Computer Vision, 2000.

[44] J. Deutscher, A. Blake and I. Reid. Articulated Body Motion Capture by Annealed Particle Filtering. In IEEE Conference on Computer Vision and Pattern Recognition, 2000.

[45] G. Mori and J. Malik. Estimating Human Body Configurations using Shape Context Matching. In IEEE Conference on Computer Vision and Pattern Recognition, 2000.

Appendix A

Stereo Camera Setup

The following describes how to calibrate and setup the SVS (Small Vision Systems) stereo camera system, which was used in our approach.

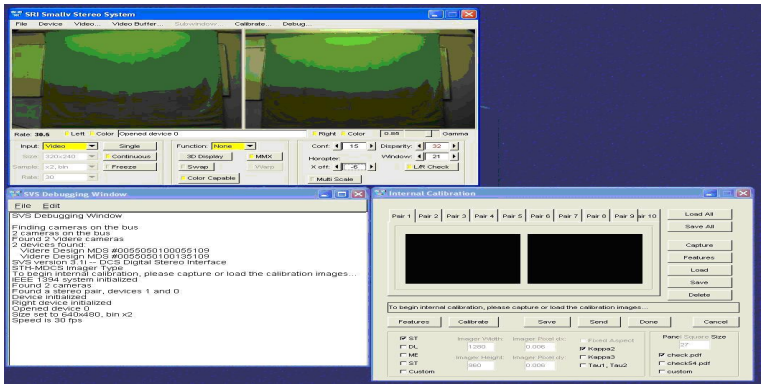

Figure 30: Stereo Calibration - Step 1

Figure 30 shows the interface to camera calibration for the SVS stereo system. The top window on the left in the figure has two images obtained from the right and left cameras respectively. For calibration, the bottom right window is used.

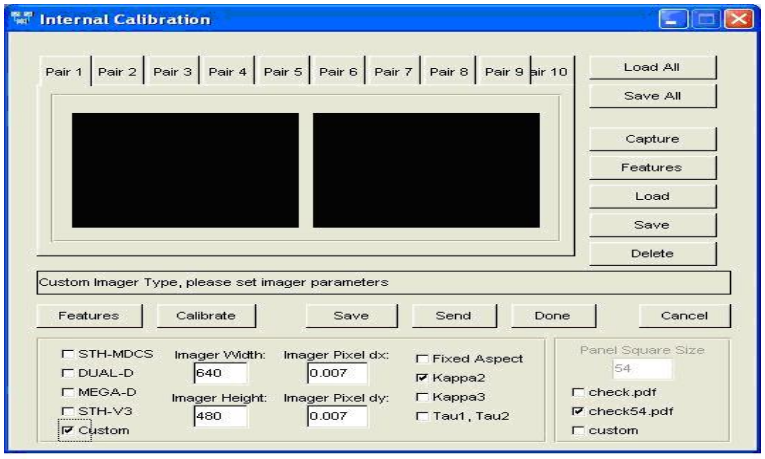

Figure 31: Stereo Calibration - Step 2

Figure 31shows the interface window used to calibrate the cameras. Ten pairs of images should be captured using a checkerboard pattern. *Features* and *Calibrate* are the two important buttons used for calibration.

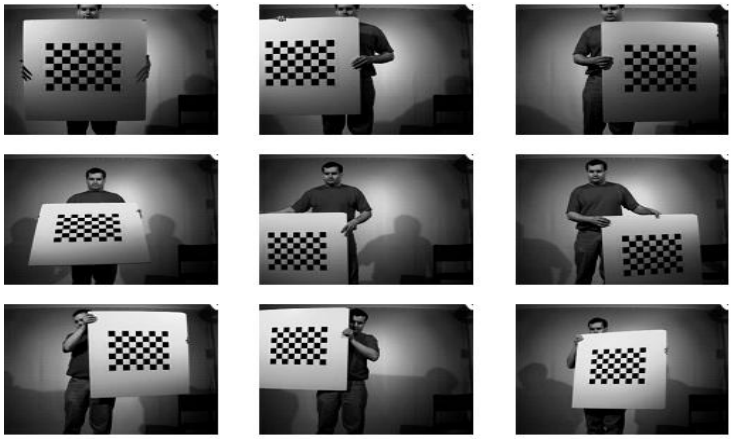

Figure 32: Stereo Calibration - Step 3

Each pair of image is taken by holding the checkerboard pattern in front of the cameras as shown in figure 32. At each time, the position and orientation of the checkerboard should be different. This is because knowing the same feature points at different angles and orientations, SVS software can calculate the exact camera parameters.

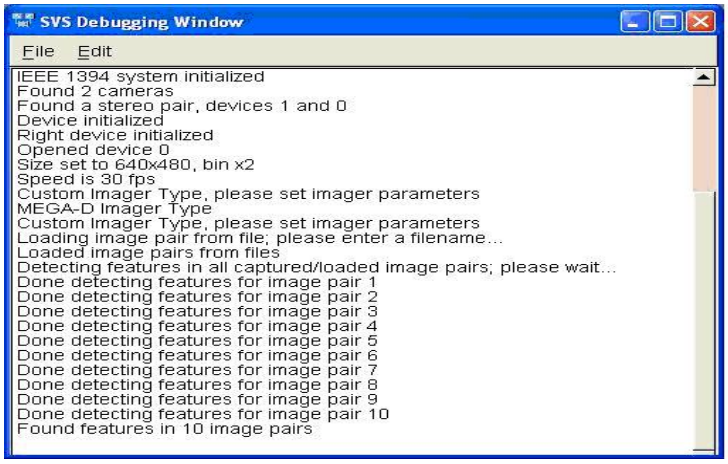

Figure 33: Stereo Calibration - Step 4

After capturing ten pairs of images, the *Features* button is selected. This will cause a window to appear as shown in figure 33. At this stage, the SVS software tries to find similar features in all the ten images. Finally, the calibrate is done by selecting the *Calibrate* button. This will calculate the camera parameters as

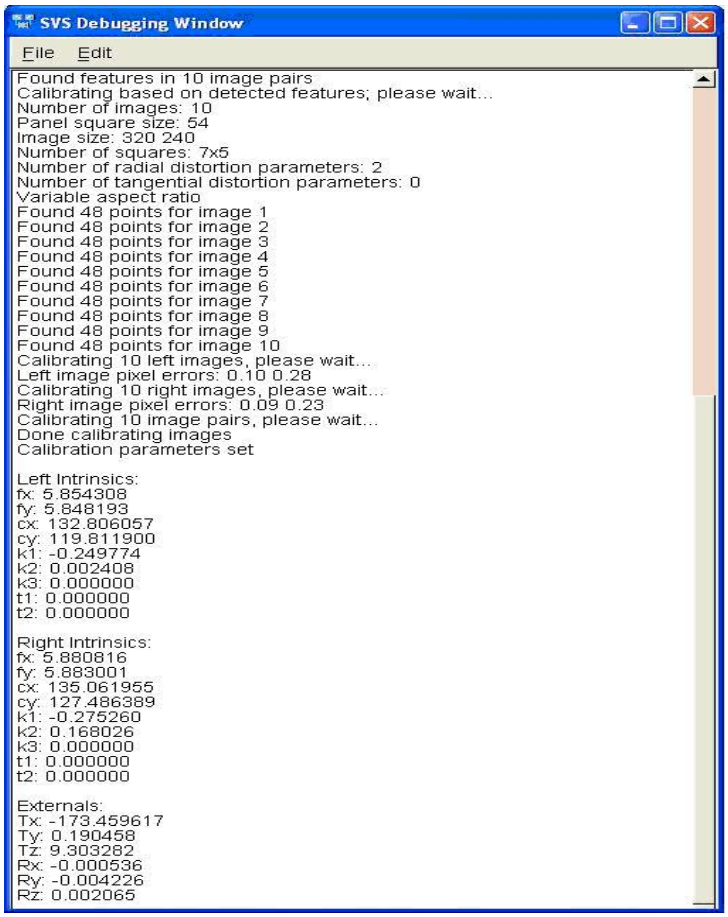

Figure 34: Stereo Calibration - Step 5

shown in figure 34. These parameters can be written to a file. They can then be used each time an application starts. If there is a slight change in camera position and orientation, calibration should be done again.